THE INVENTION CLAIMED IS:

1. A scheduler for a network processor, comprising:

one or more scheduling queues, each adapted to define a respective sequence in which flows are to be serviced; and

one or more empty indicators, each empty indicator being associated with a respective scheduling queue to indicate whether the respective scheduling queue is empty.

2. The scheduler of claim 1, wherein each empty indicator is a bit in a register.

3. The scheduler of claim 1, wherein the one or more scheduling queues include 64 scheduling queues and the one or more empty indicators include 64 empty indicators.

4. The scheduler of claim 1, wherein each scheduling queue includes 512 slots to which flows are attachable.

5. A method of dequeuing a flow from a scheduling queue, comprising:

examining an empty indicator associated with the scheduling queue;

refraining from searching the scheduling queue if the empty indicator indicates that the scheduling queue is empty;

searching the scheduling queue if the empty indicator indicates that the scheduling queue is not empty; and

detaching from the scheduling queue a winning flow found in the searching step.

6. The method of claim 5, further comprising, prior to the examining step, selecting the scheduling queue from among a plurality of scheduling queues in a round robin process.

7. The method of claim 5, wherein the searching step includes searching a plurality of subqueues included in the scheduling queue, the subqueues having mutually different respective ranges and resolutions.

8. The method of claim 5, wherein the examining step includes checking a bit in a register.

9. A method of enqueuing a flow to a scheduling queue, comprising:

attaching a flow to the scheduling queue; and

placing an empty indicator associated with the scheduling queue in a condition to indicate that the scheduling queue is not empty.

10. The method of claim 9, wherein the attaching step includes assigning the flow to a slot in the scheduling queue according to the formula CP + ((WF x FS)/SF), where:

CP is a pointer that indicates a current position in the scheduling queue;

WF is a weighting factor associated with the flow;

FS is a size of a data frame associated with the flow; and

SF is a scaling factor.

11. The method of claim 9, wherein the placing step includes setting a bit in a register.

12. The method of claim 9, wherein the placing step includes resetting a bit in a register.

13. A method of dequeuing a flow from a scheduling queue, comprising:

examining an empty indicator associated with the scheduling queue;

refraining from searching the scheduling queue if the empty indicator indicates that the scheduling queue is empty;

searching the scheduling queue if the empty indicator indicates that the scheduling queue is not empty;

if a winning flow is found in the searching step, detaching the winning flow from the scheduling queue; and

if no flow is found in the searching step, placing the empty indicator in a condition to indicate that the scheduling queue is empty.

14. The method of claim 13, further comprising, prior to the examining step, selecting the scheduling queue from among a plurality of scheduling queues in a round robin process.

15. The method of claim 13, wherein the searching step includes searching a plurality of subqueues included in

the scheduling queue, the subqueues having mutually different respective ranges and resolutions.

16. The method of claim 13, wherein the examining step includes checking a bit in a register.

17. The method of claim 13, wherein, if the detaching step is performed, a further search of the scheduling queue is performed to determine whether any flows are enqueued in the scheduling queue other than the flow detached in the detaching step.

18. The method of claim 17, wherein the empty indicator is placed in a condition to indicate that the scheduling queue is empty if the further search of the scheduling queue determines that there are no flows in the scheduling queue other than the flow detached in the detaching step.

19. A scheduler for a network processor, comprising:

one or more scheduling queues, each adapted to define a respective sequence in which flows are to be serviced; and

one or more empty indicators, each empty indicator being associated with a respective scheduling queue to indicate whether the respective scheduling queue is empty;

wherein the scheduler is adapted to:

examine an empty indicator associated with a first scheduling queue;

refrain from searching the first scheduling queue if the empty indicator indicates that the first scheduling queue is empty;

search the first scheduling queue if the empty indicator indicates that the first scheduling queue is not empty; and

detach from the first scheduling queue a winning flow found in the search of the first scheduling queue.

20. A scheduler for a network processor, comprising:

one or more scheduling queues, each adapted to define a respective sequence in which flows are to be serviced; and

one or more empty indicators, each empty indicator being associated with a respective scheduling queue to indicate whether the respective scheduling queue is empty;

wherein the scheduler is adapted to:

attach a flow to a first scheduling queue; and

place an empty indicator associated with the first scheduling queue in a condition to indicate that the first scheduling queue is not empty.

21. A scheduler for a network processor, comprising:

one or more scheduling queues, each adapted to define a respective sequence in which flows are to be serviced; and

one or more empty indicators, each empty indicator being associated with a respective scheduling queue to indicate whether the respective scheduling queue is empty;

wherein the scheduler is adapted to:

examine an empty indicator associated with a first scheduling queue;

refrain from searching the first scheduling queue if the empty indicator indicates that the first scheduling queue is empty;

search the first scheduling queue if the empty indicator indicates that the first scheduling queue is not empty;

if a winning flow is found by the search of the first scheduling queue, detach the winning flow from the first scheduling queue; and

if no flow is found by the search of the first scheduling queue, place the empty indicator in a condition to indicate that the first scheduling queue is empty.

22. A computer program product adapted to dequeue a flow from a scheduling queue, the computer program product comprising:

a medium readable by a computer, the computer readable medium having computer program code adapted to:

examine an empty indicator associated with the scheduling queue;

refrain from searching the scheduling queue if the empty indicator indicates that the scheduling queue is empty;

search the scheduling queue if the empty indicator indicates that the scheduling queue is not empty; and

detach from the scheduling queue a winning flow found in the search of the scheduling queue.

23. A computer program product adapted to enqueue a flow to a scheduling queue, the computer program product comprising:

a medium readable by a computer, the computer readable medium having computer program code adapted to:

attach a flow to the scheduling queue; and

place an empty indicator associated with the scheduling queue in a condition to indicate that the scheduling queue is not empty.

24. A computer program product adapted to dequeue a flow from a scheduling queue, the computer program product comprising:

a medium readable by a computer, the computer readable medium having computer program code adapted to:

examine an empty indicator associated with the scheduling queue;

refrain from searching the scheduling queue if the empty indicator indicates that the scheduling queue is empty;

search the scheduling queue if the empty indicator indicates that the scheduling queue is not empty;

if a winning flow is found by the search of the scheduling queue, detach the winning flow from the scheduling queue; and

if no flow is found by the search of the scheduling queue, place the empty indicator in a condition to indicate that the scheduling queue is empty.